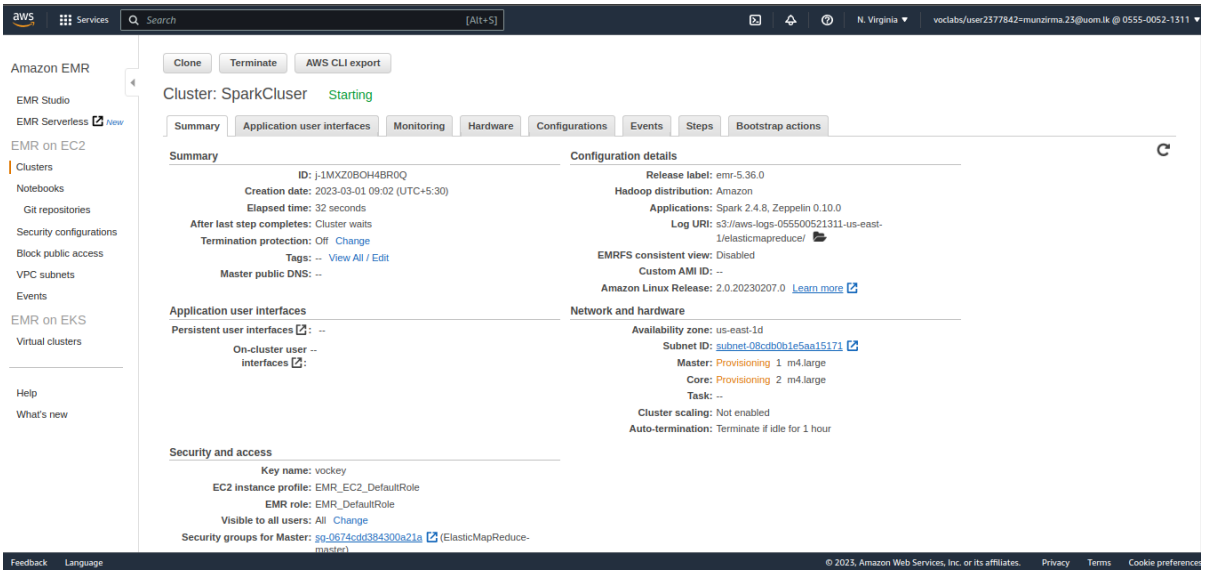
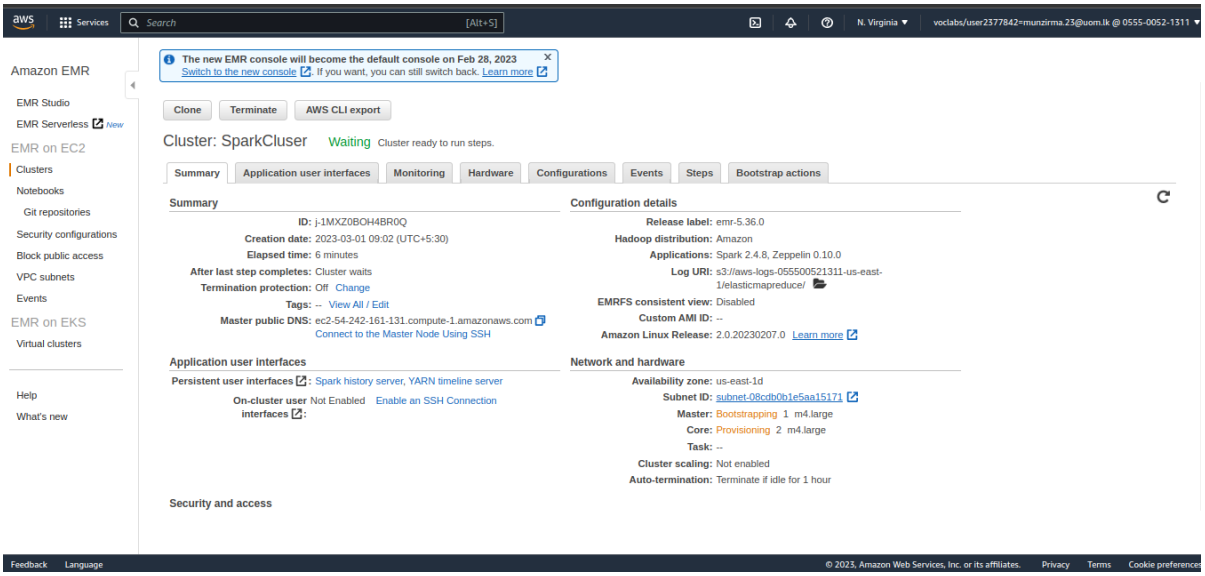
## Spark

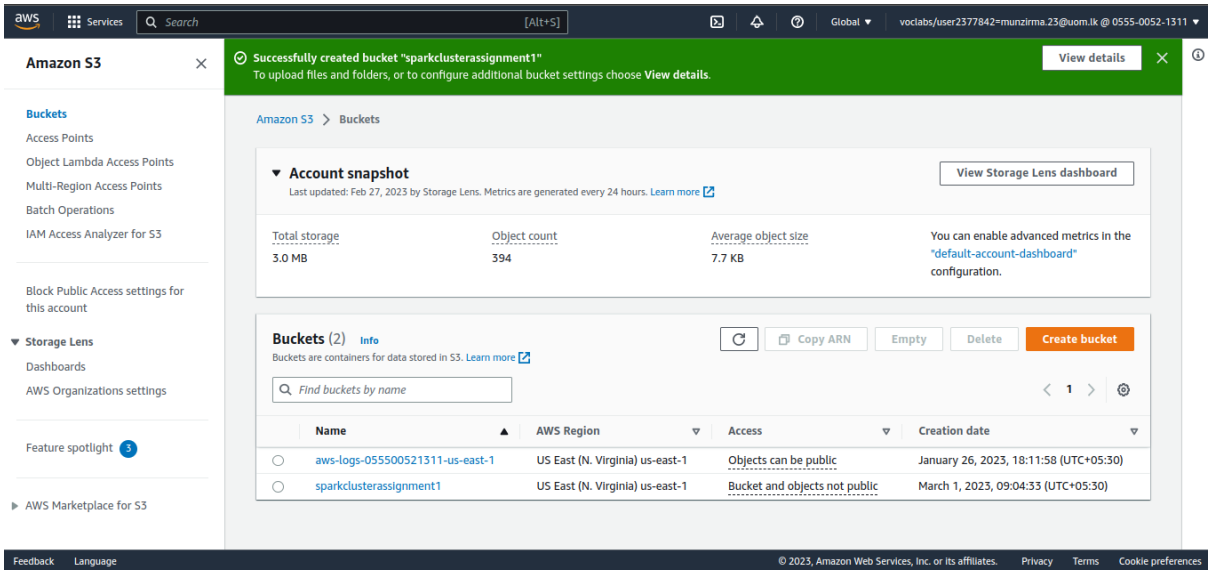
## Screen Shots of the analysis

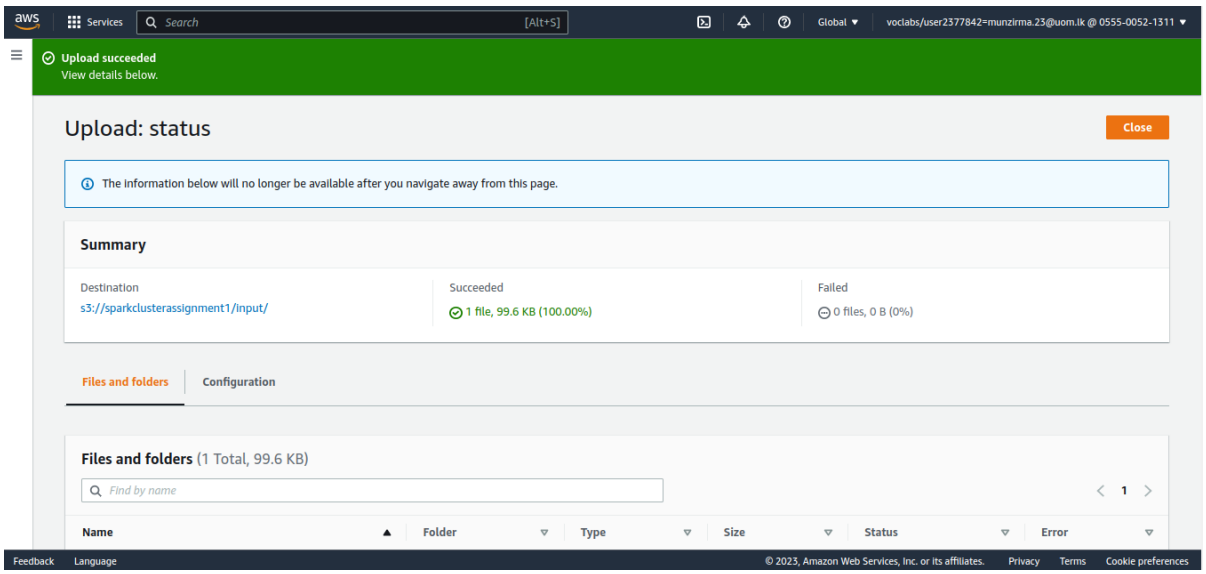
1. Launching an Amazon EMR cluster



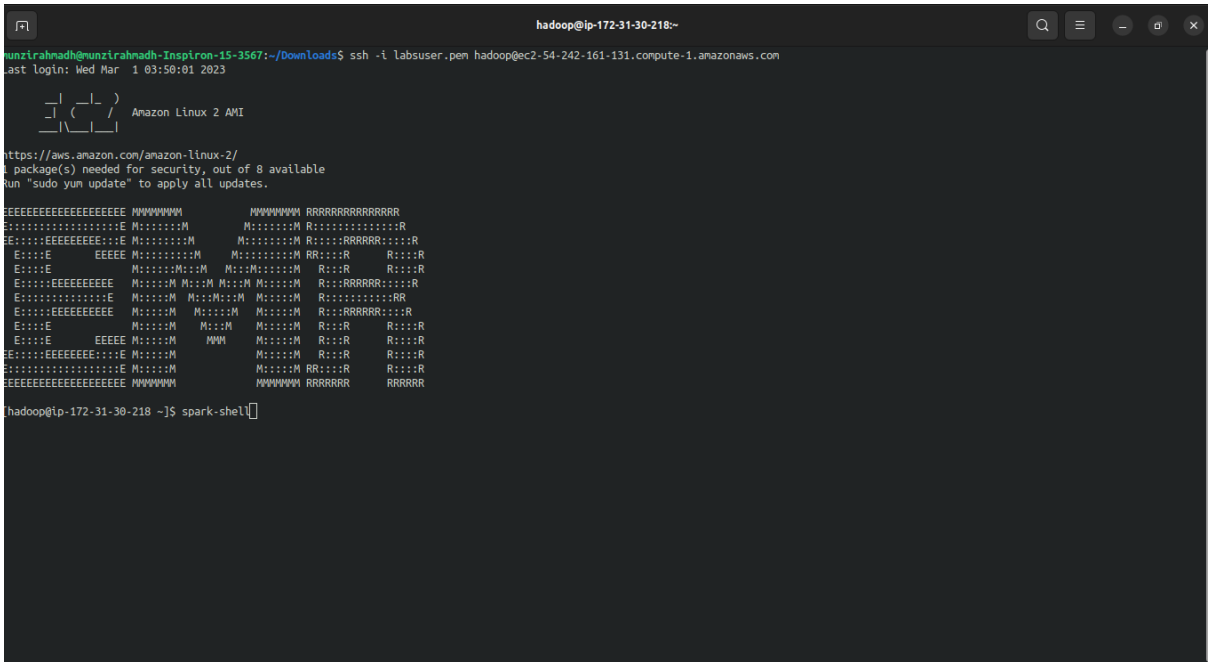



2. Create an S3 Bucket and Upload data and CSV file to S3 bucket



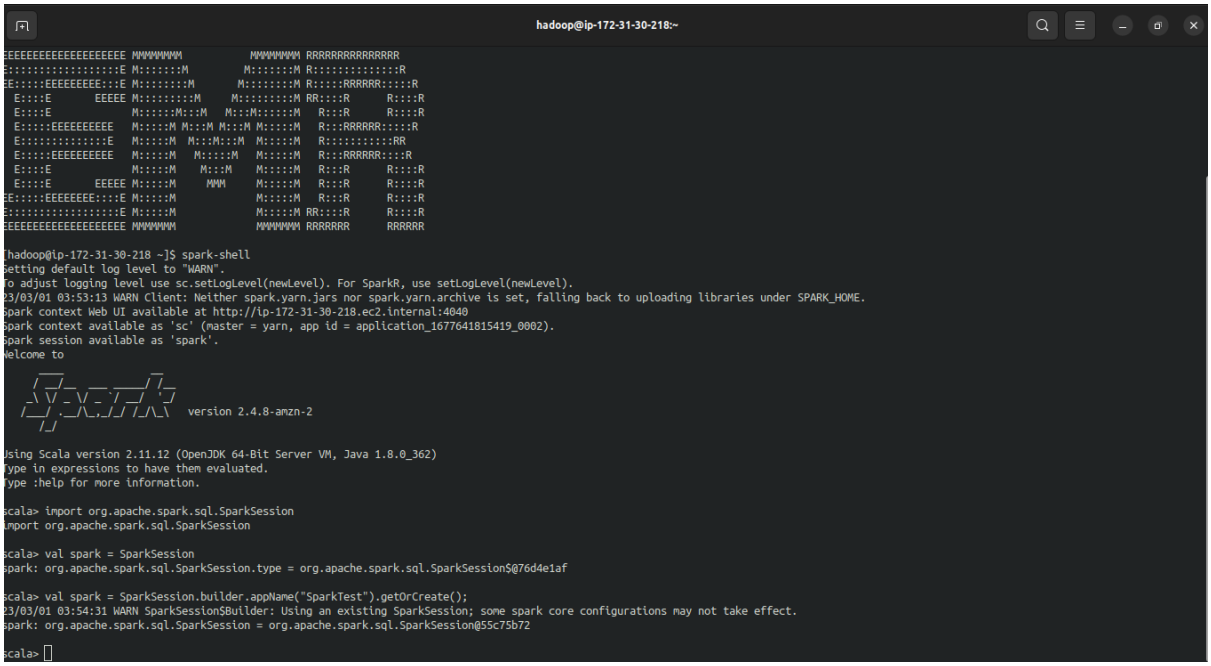



3. Connecting to the Hadoop main node by using SSH



4. Running spark

5. Imports and initializing the spark.



```
hadoop@ip-172-31-30-218:~
EEEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE::::::EEEEEEEEE:::E M::::::::M       M::::::::M R::::::RRRRRR:::::R
  E::::::E       EEEEE M:::::::::M     M:::::::::M RR:::::R     R::::R
  E::::::E             M::::::M:::M   M:::M::::::M   R:::R     R::::R
  E::::::EEEEEEEEEE    M::::::M M:::M M::: M::::::M   R:::RRRRRR::::::R
  E::::::::::::::::E   M::::::M  M:::M:::M  M::::::M   R::::::::::::RR
  E::::::EEEEEEEEEE    M::::::M   M:::::M   M::::::M   R:::RRRRRR::::R
  E::::::E             M::::::M    M:::M    M::::::M   R:::R     R::::R
  E::::::E       EEEEE M::::::M     MMM     M::::::M   R:::R     R::::R
EE::::::EEEEEEEE:::E M::::::M             M::::::M   R:::R     R::::R
E::::::::::::::::::E M::::::M             M::::::M RR:::::R     R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM              MMMMMMM RRRRRRR     RRRRRR

[hadoop@ip-172-31-30-218 ~]$ spark-shell
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
23/03/01 03:53:13 WARN Client: Neither spark.yarn.jars nor spark.yarn.archive is set, falling back to uploading libraries under SPARK_HOME.
Spark context Web UI available at http://ip-172-31-30-218.ec2.internal:4040
Spark context available as 'sc' (master = yarn, app id = application_1677641815419_0002).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 2.4.8-amzn-2
      /_/

Using Scala version 2.11.12 (OpenJDK 64-Bit Server VM, Java 1.8.0_362)
Type in expressions to have them evaluated.
Type :help for more information.

scala> import org.apache.spark.sql.SparkSession
import org.apache.spark.sql.SparkSession

scala> val spark = SparkSession
spark: org.apache.spark.sql.SparkSession.type = org.apache.spark.sql.SparkSession$@76d4e1af

scala> val spark = SparkSession.builder.appName("SparkTest").getOrCreate();
23/03/01 03:54:31 WARN SparkSession$Builder: Using an existing SparkSession; some spark core configurations may not take effect.
spark: org.apache.spark.sql.SparkSession = org.apache.spark.sql.SparkSession@55c75b72

scala>
```

6. Load data and create a view

```
hadoop@ip-172-31-30-218:~

  E::::E       EEEEE M:::::M     MMM     M:::::M   R:::R      R::::R
EE:::::EEEEEEEE::::E M:::::M             M:::::M   R:::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR

[hadoop@ip-172-31-30-218 ~]$ spark-shell
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
23/03/01 03:53:13 WARN Client: Neither spark.yarn.jars nor spark.yarn.archive is set, falling back to uploading libraries under SPARK_HOME.
Spark context Web UI available at http://ip-172-31-30-218.ec2.internal:4040
Spark context available as 'sc' (master = yarn, app id = application_1677641815419_0002).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 2.4.8-amzn-2
      /_/

Using Scala version 2.11.12 (OpenJDK 64-Bit Server VM, Java 1.8.0_362)
Type in expressions to have them evaluated.
Type :help for more information.

scala> import org.apache.spark.sql.SparkSession
import org.apache.spark.sql.SparkSession

scala> val spark = SparkSession
spark: org.apache.spark.sql.SparkSession.type = org.apache.spark.sql.SparkSession$@76d4e1af

scala> val spark = SparkSession.builder.appName("SparkTest").getOrCreate();
23/03/01 03:54:31 WARN SparkSession$Builder: Using an existing SparkSession; some spark core configurations may not take effect.
spark: org.apache.spark.sql.SparkSession = org.apache.spark.sql.SparkSession@55c75b72

scala> val csvFile = "s3://sparkclusterassignment1/input/DelayedFlights-updated.csv"
csvFile: String = s3://sparkclusterassignment1/input/DelayedFlights-updated.csv

scala> val df = spark.read.format("csv").option("inferSchema", "true").option("header","true").load(csvFile);
df: org.apache.spark.sql.DataFrame = [_c0: int, Year: int ... 28 more fields]

scala> df.createOrReplaceTempView("delay_flights");
23/03/01 03:56:39 WARN Utils: Truncated the string representation of a plan since it was too large. This behavior can be adjusted by setting 'spark.debug.maxToStringFields' in SparkEnv.conf.

scala>
```

7. Querying and analyzing the resulting dataset

```
csvFile: String = s3://sparkclusterassignment1/input/DelayedFlights-Updated.csv

scala> val df = spark.read.format("csv").option("inferSchema", "true").option("header","true").load(csvFile);
df: org.apache.spark.sql.DataFrame = [_c0: int, Year: int ... 28 more fields]

scala> df.createOrReplaceTempView("delay_flights");
23/03/01 03:56:39 WARN Utils: Truncated the string representation of a plan since it was too large. This behavior can be adjusted by setting 'spark.debug.maxToStringFields' in SparkEnv.conf.

scala> spark.time(spark.sql("SELECT Year, avg((CarrierDelay /ArrDelay)*100) from delay_flights GROUP BY Year").show())
23/03/01 03:57:30 WARN ObjectStore: Failed to get database global_temp, returning NoSuchObjectException
+----+-----------------------------------------------------------------------------------+
|Year|avg(((CAST(CarrierDelay AS DOUBLE) / CAST(ArrDelay AS DOUBLE)) * CAST(100 AS DOUBLE)))|
+----+-----------------------------------------------------------------------------------+
|2003|                                                                 24.557549755575373|
|2007|                                                                 19.850007017971283|
|2010|                                                                  21.89310246015957|
|2006|                                                                 30.453296261292596|
|2004|                                                                  43.64459443230066|
|2005|                                                                  28.01977637202288|
|2009|                                                                  28.33058554239575|
|2008|                                                                  28.88346981456985|
+----+-----------------------------------------------------------------------------------+
```

## 8. Other results

```
scala> spark.time(spark.sql("SELECT Year, avg((CarrierDelay /ArrDelay)*100) from delay_flights WHERE Year is NOT NULL GROUP BY Year").show())
+----+-----------------------------------------------------------------------------------------+
|Year|avg(((CAST(CarrierDelay AS DOUBLE) / CAST(ArrDelay AS DOUBLE)) * CAST(100 AS DOUBLE)))|
+----+-----------------------------------------------------------------------------------------+
|2003|                                                                     24.557549755575373|
|2007|                                                                     19.850007017971283|
|2010|                                                                      21.89310246015957|
|2006|                                                                     30.453296261292596|
|2004|                                                                      43.64459443230066|
|2005|                                                                      28.01977637202288|
|2009|                                                                      28.33058554239575|
|2008|                                                                      28.88346981456985|
+----+-----------------------------------------------------------------------------------------+

Time taken: 863 ms
```

```
scala> spark.time(spark.sql("SELECT Year, avg((NASDelay /ArrDelay)*100) from delay_flights WHERE Year is NOT NULL GROUP BY Year").show())
+----+-----------------------------------------------------------------------------------------+
|Year|avg(((CAST(NASDelay AS DOUBLE) / CAST(ArrDelay AS DOUBLE)) * CAST(100 AS DOUBLE)))|
+----+-----------------------------------------------------------------------------------------+
|2003|                                                                    29.686276314267346|
|2007|                                                                    30.625925917941924|
|2010|                                                                     33.87351363404217|
|2006|                                                                    18.119312329937703|
|2004|                                                                     18.24570061769958|
|2005|                                                                     16.63868805373129|
|2009|                                                                     37.63093330628511|
|2008|                                                                     30.16552562594132|
+----+-----------------------------------------------------------------------------------------+

Time taken: 832 ms
```

```
scala> spark.time(spark.sql("SELECT Year, avg((WeatherDelay /ArrDelay)*100) from delay_flights WHERE Year is NOT NULL GROUP BY Year").show())
+----+-----------------------------------------------------------------------------------------+
|Year|avg(((CAST(WeatherDelay AS DOUBLE) / CAST(ArrDelay AS DOUBLE)) * CAST(100 AS DOUBLE)))|
+----+-----------------------------------------------------------------------------------------+
|2003|                                                                     7.8319479664511205|
|2007|                                                                      4.042975783210287|
|2010|                                                                     2.9023312955584664|
|2006|                                                                      4.588604183967953|
|2004|                                                                     6.447527997691655 5|
|2005|                                                                       5.85069715149616|
|2009|                                                                    0.45316615137982363|
|2008|                                                                     3.7254490054008955|
+----+-----------------------------------------------------------------------------------------+

Time taken: 814 ms
```

```
scala> spark.time(spark.sql("SELECT Year, avg((LateAircraftDelay /ArrDelay)*100) from delay_flights WHERE Year is NOT NULL GROUP BY Year").show())
+----+-----------------------------------------------------------------------------------------+
|Year|avg(((CAST(LateAircraftDelay AS DOUBLE) / CAST(ArrDelay AS DOUBLE)) * CAST(100 AS DOUBLE)))|
+----+-----------------------------------------------------------------------------------------+
|2003|                                                                      37.924225963706164|
|2007|                                                                      45.252432744291134|
|2010|                                                                      41.331052610239794|
|2006|                                                                      46.838787224801735|
|2004|                                                                      31.662176952308105|
|2005|                                                                      49.490838422749654|
|2009|                                                                      33.585314999939314|
|2008|                                                                       37.22555555408794|
+----+-----------------------------------------------------------------------------------------+

Time taken: 977 ms
```

```
scala> spark.time(spark.sql("SELECT Year, avg((SecurityDelay /ArrDelay)*100) from delay_flights WHERE Year is NOT NULL GROUP BY Year").show())
+----+-----------------------------------------------------------------------------------------+
|Year|avg(((CAST(SecurityDelay AS DOUBLE) / CAST(ArrDelay AS DOUBLE)) * CAST(100 AS DOUBLE)))|
+----+-----------------------------------------------------------------------------------------+
|2003|                                                                                      0.0|
|2007|                                                                    0.228658536585536586|
|2010|                                                                                      0.0|
|2006|                                                                                      0.0|
|2004|                                                                                      0.0|
|2005|                                                                                      0.0|
|2009|                                                                                      0.0|
|2008|                                                                                      0.0|
+----+-----------------------------------------------------------------------------------------+

Time taken: 787 ms
```

Year wise carrier delay from 2003-2010

```
scala> spark.time(spark.sql("SELECT Year, avg((CarrierDelay /ArrDelay)*100) from delay_flights WHERE Year is NOT NULL GROUP BY Year").show())
+----+------------------------------------------------------------------------------------+
|Year|avg(((CAST(CarrierDelay AS DOUBLE) / CAST(ArrDelay AS DOUBLE)) * CAST(100 AS DOUBLE)))|
+----+------------------------------------------------------------------------------------+
|2003|                                                                  24.557549755575373|
|2007|                                                                  19.850007017971283|
|2010|                                                                   21.89310246015957|
|2006|                                                                  30.453296261292596|
|2004|                                                                   43.64459443230066|
|2005|                                                                   28.01977637202288|
|2009|                                                                   28.33058554239575|
|2008|                                                                   28.88346981456985|
+----+------------------------------------------------------------------------------------+

Time taken: 732 ms
```

```
scala> spark.time(spark.sql("SELECT Year, avg((CarrierDelay /ArrDelay)*100) from delay_flights WHERE Year is NOT NULL GROUP BY Year").show())
+----+------------------------------------------------------------------------------------+
|Year|avg(((CAST(CarrierDelay AS DOUBLE) / CAST(ArrDelay AS DOUBLE)) * CAST(100 AS DOUBLE)))|
+----+------------------------------------------------------------------------------------+
|2003|                                                                  24.557549755575373|
|2007|                                                                  19.850007017971283|
|2010|                                                                   21.89310246015957|
|2006|                                                                  30.453296261292596|
|2004|                                                                   43.64459443230066|
|2005|                                                                   28.01977637202288|
|2009|                                                                   28.33058554239575|
|2008|                                                                   28.88346981456985|
+----+------------------------------------------------------------------------------------+

Time taken: 662 ms

scala>
```

```
scala> spark.time(spark.sql("SELECT Year, avg((CarrierDelay /ArrDelay)*100) from delay_flights WHERE Year is NOT NULL GROUP BY Year").show())
+----+------------------------------------------------------------------------------------+
|Year|avg(((CAST(CarrierDelay AS DOUBLE) / CAST(ArrDelay AS DOUBLE)) * CAST(100 AS DOUBLE)))|
+----+------------------------------------------------------------------------------------+
|2003|                                                                  24.557549755575373|
|2007|                                                                  19.850007017971283|
|2010|                                                                   21.89310246015957|
|2006|                                                                  30.453296261292596|
|2004|                                                                   43.64459443230066|
|2005|                                                                   28.01977637202288|
|2009|                                                                   28.33058554239575|
|2008|                                                                   28.88346981456985|
+----+------------------------------------------------------------------------------------+

Time taken: 808 ms
```

```
scala> spark.time(spark.sql("SELECT Year, avg((CarrierDelay /ArrDelay)*100) from delay_flights WHERE Year is NOT NULL GROUP BY Year").show())
+----+------------------------------------------------------------------------------------+
|Year|avg(((CAST(CarrierDelay AS DOUBLE) / CAST(ArrDelay AS DOUBLE)) * CAST(100 AS DOUBLE)))|
+----+------------------------------------------------------------------------------------+
|2003|                                                                  24.557549755575373|
|2007|                                                                  19.850007017971283|
|2010|                                                                   21.89310246015957|
|2006|                                                                  30.453296261292596|
|2004|                                                                   43.64459443230066|
|2005|                                                                   28.01977637202288|
|2009|                                                                   28.33058554239575|
|2008|                                                                   28.88346981456985|
+----+------------------------------------------------------------------------------------+

Time taken: 643 ms
```

```
scala> spark.time(spark.sql("SELECT Year, avg((CarrierDelay /ArrDelay)*100) from delay_flights WHERE Year is NOT NULL GROUP BY Year").show())
+----+------------------------------------------------------------------------------------+
|Year|avg(((CAST(CarrierDelay AS DOUBLE) / CAST(ArrDelay AS DOUBLE)) * CAST(100 AS DOUBLE)))|
+----+------------------------------------------------------------------------------------+
|2003|                                                                  24.557549755575373|
|2007|                                                                  19.850007017971283|
|2010|                                                                   21.89310246015957|
|2006|                                                                  30.453296261292596|
|2004|                                                                   43.64459443230066|
|2005|                                                                   28.01977637202288|
|2009|                                                                   28.33058554239575|
|2008|                                                                   28.88346981456985|
+----+------------------------------------------------------------------------------------+

Time taken: 834 ms

scala>
```

# Year wise NAS delay from 2003-2010

```
scala> spark.time(spark.sql("SELECT Year, avg((NASDelay /ArrDelay)*100) from delay_flights WHERE Year is NOT NULL GROUP BY Year").show());
+----+------------------------------------------------------------------------------------+
|Year|avg(((CAST(NASDelay AS DOUBLE) / CAST(ArrDelay AS DOUBLE)) * CAST(100 AS DOUBLE)))|
+----+------------------------------------------------------------------------------------+
|2003|                                                                  29.686276314267346|
|2007|                                                                  30.625925917941924|
|2010|                                                                   33.87351363404217|
|2006|                                                                  18.119312329937703|
|2004|                                                                   18.24570061769958|
|2005|                                                                   16.63868805373129|
|2009|                                                                   37.63093330628511|
|2008|                                                                   30.16552562594132|
+----+------------------------------------------------------------------------------------+

Time taken: 813 ms
```

```
scala> spark.time(spark.sql("SELECT Year, avg((NASDelay /ArrDelay)*100) from delay_flights WHERE Year is NOT NULL GROUP BY Year").show());
+----+-----------------------------------------------------------------------------+
|Year|avg(((CAST(NASDelay AS DOUBLE) / CAST(ArrDelay AS DOUBLE)) * CAST(100 AS DOUBLE)))|
+----+-----------------------------------------------------------------------------+
|2003|                                                                29.686276314267346|
|2007|                                                                30.625925917941924|
|2010|                                                                 33.87351363404217|
|2006|                                                                18.119312329937703|
|2004|                                                                 18.24570061769958|
|2005|                                                                 16.63868805373129|
|2009|                                                                 37.63093330628511|
|2008|                                                                 30.16552562594132|
+----+-----------------------------------------------------------------------------+

Time taken: 764 ms
```

```
scala> spark.time(spark.sql("SELECT Year, avg((NASDelay /ArrDelay)*100) from delay_flights WHERE Year is NOT NULL GROUP BY Year").show());
+----+-----------------------------------------------------------------------------+
|Year|avg(((CAST(NASDelay AS DOUBLE) / CAST(ArrDelay AS DOUBLE)) * CAST(100 AS DOUBLE)))|
+----+-----------------------------------------------------------------------------+
|2003|                                                                29.686276314267346|
|2007|                                                                30.625925917941924|
|2010|                                                                 33.87351363404217|
|2006|                                                                18.119312329937703|
|2004|                                                                 18.24570061769958|
|2005|                                                                 16.63868805373129|
|2009|                                                                 37.63093330628511|
|2008|                                                                 30.16552562594132|
+----+-----------------------------------------------------------------------------+

Time taken: 610 ms
```

```
scala> spark.time(spark.sql("SELECT Year, avg((NASDelay /ArrDelay)*100) from delay_flights WHERE Year is NOT NULL GROUP BY Year").show());
+----+-----------------------------------------------------------------------------+
|Year|avg(((CAST(NASDelay AS DOUBLE) / CAST(ArrDelay AS DOUBLE)) * CAST(100 AS DOUBLE)))|
+----+-----------------------------------------------------------------------------+
|2003|                                                                29.686276314267346|
|2007|                                                                30.625925917941924|
|2010|                                                                 33.87351363404217|
|2006|                                                                18.119312329937703|
|2004|                                                                 18.24570061769958|
|2005|                                                                 16.63868805373129|
|2009|                                                                 37.63093330628511|
|2008|                                                                 30.16552562594132|
+----+-----------------------------------------------------------------------------+

Time taken: 815 ms
```

```
scala> spark.time(spark.sql("SELECT Year, avg((NASDelay /ArrDelay)*100) from delay_flights WHERE Year is NOT NULL GROUP BY Year").show());
+----+-----------------------------------------------------------------------------+
|Year|avg(((CAST(NASDelay AS DOUBLE) / CAST(ArrDelay AS DOUBLE)) * CAST(100 AS DOUBLE)))|
+----+-----------------------------------------------------------------------------+
|2003|                                                                29.686276314267346|
|2007|                                                                30.625925917941924|
|2010|                                                                 33.87351363404217|
|2006|                                                                18.119312329937703|
|2004|                                                                 18.24570061769958|
|2005|                                                                 16.63868805373129|
|2009|                                                                 37.63093330628511|
|2008|                                                                 30.16552562594132|
+----+-----------------------------------------------------------------------------+

Time taken: 590 ms
```

# Year wise Weather delay from 2003-2010

```
scala> spark.time(spark.sql("SELECT Year, avg((WeatherDelay /ArrDelay)*100) from delay_flights WHERE Year is NOT NULL GROUP BY Year").show());
+----+-----------------------------------------------------------------------------+
|Year|avg(((CAST(WeatherDelay AS DOUBLE) / CAST(ArrDelay AS DOUBLE)) * CAST(100 AS DOUBLE)))|
+----+-----------------------------------------------------------------------------+
|2003|                                                                 7.8319479664511205|
|2007|                                                                  4.042975783210287|
|2010|                                                                 2.9023312955584664|
|2006|                                                                  4.588604183967953|
|2004|                                                                 6.4475279976916555|
|2005|                                                                   5.85069715149616|
|2009|                                                                0.45316615137982363|
|2008|                                                                 3.7254490054008955|
+----+-----------------------------------------------------------------------------+

Time taken: 773 ms
```

```
scala> spark.time(spark.sql("SELECT Year, avg((WeatherDelay /ArrDelay)*100) from delay_flights WHERE Year is NOT NULL GROUP BY Year").show());
+----+-----------------------------------------------------------------------------+
|Year|avg(((CAST(WeatherDelay AS DOUBLE) / CAST(ArrDelay AS DOUBLE)) * CAST(100 AS DOUBLE)))|
+----+-----------------------------------------------------------------------------+
|2003|                                                                 7.8319479664511205|
|2007|                                                                  4.042975783210287|
|2010|                                                                 2.9023312955584664|
|2006|                                                                  4.588604183967953|
|2004|                                                                 6.4475279976916555|
|2005|                                                                   5.85069715149616|
|2009|                                                                0.45316615137982363|
|2008|                                                                 3.7254490054008955|
+----+-----------------------------------------------------------------------------+

Time taken: 689 ms
```

```
scala> spark.time(spark.sql("SELECT Year, avg((WeatherDelay /ArrDelay)*100) from delay_flights WHERE Year is NOT NULL GROUP BY Year").show());
+----+-------------------------------------------------------------------------------------+
|Year|avg(((CAST(WeatherDelay AS DOUBLE) / CAST(ArrDelay AS DOUBLE)) * CAST(100 AS DOUBLE)))|
+----+-------------------------------------------------------------------------------------+
|2003|                                                                   7.8319479664511205|
|2007|                                                                    4.042975783210287|
|2010|                                                                   2.9023312955584664|
|2006|                                                                    4.588604183967953|
|2004|                                                                   6.4475279976916555|
|2005|                                                                     5.85069715149616|
|2009|                                                                  0.45316615137982363|
|2008|                                                                   3.7254490054008955|
+----+-------------------------------------------------------------------------------------+

Time taken: 735 ms
```

```
scala> spark.time(spark.sql("SELECT Year, avg((WeatherDelay /ArrDelay)*100) from delay_flights WHERE Year is NOT NULL GROUP BY Year").show());
+----+-------------------------------------------------------------------------------------+
|Year|avg(((CAST(WeatherDelay AS DOUBLE) / CAST(ArrDelay AS DOUBLE)) * CAST(100 AS DOUBLE)))|
+----+-------------------------------------------------------------------------------------+
|2003|                                                                   7.8319479664511205|
|2007|                                                                    4.042975783210287|
|2010|                                                                   2.9023312955584664|
|2006|                                                                    4.588604183967953|
|2004|                                                                   6.4475279976916555|
|2005|                                                                     5.85069715149616|
|2009|                                                                  0.45316615137982363|
|2008|                                                                   3.7254490054008955|
+----+-------------------------------------------------------------------------------------+

Time taken: 743 ms
```

```
scala> spark.time(spark.sql("SELECT Year, avg((WeatherDelay /ArrDelay)*100) from delay_flights WHERE Year is NOT NULL GROUP BY Year").show());
+----+-------------------------------------------------------------------------------------+
|Year|avg(((CAST(WeatherDelay AS DOUBLE) / CAST(ArrDelay AS DOUBLE)) * CAST(100 AS DOUBLE)))|
+----+-------------------------------------------------------------------------------------+
|2003|                                                                   7.8319479664511205|
|2007|                                                                    4.042975783210287|
|2010|                                                                   2.9023312955584664|
|2006|                                                                    4.588604183967953|
|2004|                                                                   6.4475279976916555|
|2005|                                                                     5.85069715149616|
|2009|                                                                  0.45316615137982363|
|2008|                                                                   3.7254490054008955|
+----+-------------------------------------------------------------------------------------+

Time taken: 771 ms

scala> 
```

# Year wise late aircraft delay from 2003-2010

```
scala> spark.time(spark.sql("SELECT Year, avg((LateAircraftDelay /ArrDelay)*100) from delay_flights WHERE Year is NOT NULL GROUP BY Year").show());
+----+-------------------------------------------------------------------------------------+
|Year|avg(((CAST(LateAircraftDelay AS DOUBLE) / CAST(ArrDelay AS DOUBLE)) * CAST(100 AS DOUBLE)))|
+----+-------------------------------------------------------------------------------------+
|2003|                                                                  37.924225963706164|
|2007|                                                                  45.252432744291134|
|2010|                                                                  41.331052610239794|
|2006|                                                                  46.838787224801735|
|2004|                                                                  31.662176952308105|
|2005|                                                                  49.490838422749654|
|2009|                                                                  33.585314999939314|
|2008|                                                                   37.22555555408794|
+----+-------------------------------------------------------------------------------------+

Time taken: 589 ms

scala> 
```

```
scala> spark.time(spark.sql("SELECT Year, avg((LateAircraftDelay /ArrDelay)*100) from delay_flights WHERE Year is NOT NULL GROUP BY Year").show());
+----+-------------------------------------------------------------------------------------+
|Year|avg(((CAST(LateAircraftDelay AS DOUBLE) / CAST(ArrDelay AS DOUBLE)) * CAST(100 AS DOUBLE)))|
+----+-------------------------------------------------------------------------------------+
|2003|                                                                  37.924225963706164|
|2007|                                                                  45.252432744291134|
|2010|                                                                  41.331052610239794|
|2006|                                                                  46.838787224801735|
|2004|                                                                  31.662176952308105|
|2005|                                                                  49.490838422749654|
|2009|                                                                  33.585314999939314|
|2008|                                                                   37.22555555408794|
+----+-------------------------------------------------------------------------------------+

Time taken: 583 ms
```

```
scala> spark.time(spark.sql("SELECT Year, avg((LateAircraftDelay /ArrDelay)*100) from delay_flights WHERE Year is NOT NULL GROUP BY Year").show());
+----+-------------------------------------------------------------------------------------+
|Year|avg(((CAST(LateAircraftDelay AS DOUBLE) / CAST(ArrDelay AS DOUBLE)) * CAST(100 AS DOUBLE)))|
+----+-------------------------------------------------------------------------------------+
|2003|                                                                  37.924225963706164|
|2007|                                                                  45.252432744291134|
|2010|                                                                  41.331052610239794|
|2006|                                                                  46.838787224801735|
|2004|                                                                  31.662176952308105|
|2005|                                                                  49.490838422749654|
|2009|                                                                  33.585314999939314|
|2008|                                                                   37.22555555408794|
+----+-------------------------------------------------------------------------------------+

Time taken: 611 ms

scala> 
```

```
scala> spark.time(spark.sql("SELECT Year, avg((LateAircraftDelay /ArrDelay)*100) from delay_flights WHERE Year is NOT NULL GROUP BY Year").show());
+----+-----------------------------------------------------------------------------------------+
|Year|avg(((CAST(LateAircraftDelay AS DOUBLE) / CAST(ArrDelay AS DOUBLE)) * CAST(100 AS DOUBLE)))|
+----+-----------------------------------------------------------------------------------------+
|2003|                                                                       37.924225963706164|
|2007|                                                                       45.252432744291134|
|2010|                                                                       41.331052610239794|
|2006|                                                                       46.838787224801735|
|2004|                                                                       31.662176952308105|
|2005|                                                                       49.490838422749654|
|2009|                                                                       33.585314999939314|
|2008|                                                                        37.22555555408794|
+----+-----------------------------------------------------------------------------------------+

Time taken: 538 ms
```

```
scala> spark.time(spark.sql("SELECT Year, avg((LateAircraftDelay /ArrDelay)*100) from delay_flights WHERE Year is NOT NULL GROUP BY Year").show());
+----+-----------------------------------------------------------------------------------------+
|Year|avg(((CAST(LateAircraftDelay AS DOUBLE) / CAST(ArrDelay AS DOUBLE)) * CAST(100 AS DOUBLE)))|
+----+-----------------------------------------------------------------------------------------+
|2003|                                                                       37.924225963706164|
|2007|                                                                       45.252432744291134|
|2010|                                                                       41.331052610239794|
|2006|                                                                       46.838787224801735|
|2004|                                                                       31.662176952308105|
|2005|                                                                       49.490838422749654|
|2009|                                                                       33.585314999939314|
|2008|                                                                        37.22555555408794|
+----+-----------------------------------------------------------------------------------------+

Time taken: 727 ms

scala> 
```

# Year wise security delay from 2003-2010

```
scala> spark.time(spark.sql("SELECT Year, avg((SecurityDelay /ArrDelay)*100) from delay_flights WHERE Year is NOT NULL GROUP BY Year").show());
+----+-----------------------------------------------------------------------------------------+
|Year|avg(((CAST(SecurityDelay AS DOUBLE) / CAST(ArrDelay AS DOUBLE)) * CAST(100 AS DOUBLE)))|
+----+-----------------------------------------------------------------------------------------+
|2003|                                                                                      0.0|
|2007|                                                                     0.22865853658536586|
|2010|                                                                                      0.0|
|2006|                                                                                      0.0|
|2004|                                                                                      0.0|
|2005|                                                                                      0.0|
|2009|                                                                                      0.0|
|2008|                                                                                      0.0|
+----+-----------------------------------------------------------------------------------------+

Time taken: 476 ms

scala> 
```

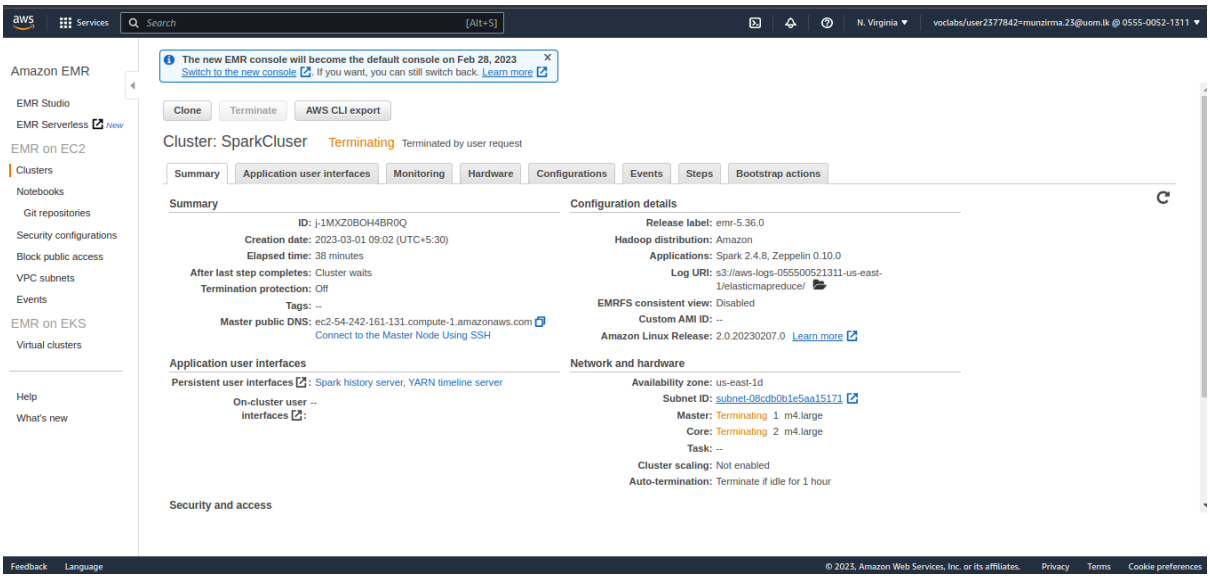
```
scala> spark.time(spark.sql("SELECT Year, avg((SecurityDelay /ArrDelay)*100) from delay_flights WHERE Year is NOT NULL GROUP BY Year").show());
+----+-----------------------------------------------------------------------------------------+
|Year|avg(((CAST(SecurityDelay AS DOUBLE) / CAST(ArrDelay AS DOUBLE)) * CAST(100 AS DOUBLE)))|
+----+-----------------------------------------------------------------------------------------+
|2003|                                                                                      0.0|
|2007|                                                                     0.22865853658536586|
|2010|                                                                                      0.0|
|2006|                                                                                      0.0|
|2004|                                                                                      0.0|
|2005|                                                                                      0.0|
|2009|                                                                                      0.0|
|2008|                                                                                      0.0|
+----+-----------------------------------------------------------------------------------------+

Time taken: 812 ms
```

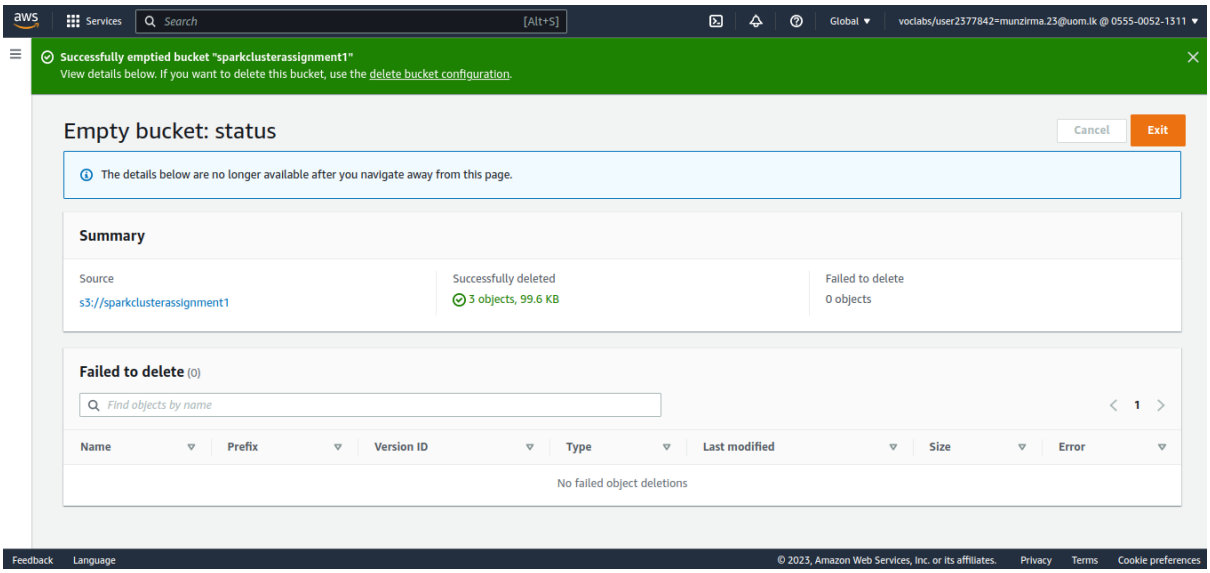
```
scala> spark.time(spark.sql("SELECT Year, avg((SecurityDelay /ArrDelay)*100) from delay_flights WHERE Year is NOT NULL GROUP BY Year").show());
+----+-----------------------------------------------------------------------------------------+
|Year|avg(((CAST(SecurityDelay AS DOUBLE) / CAST(ArrDelay AS DOUBLE)) * CAST(100 AS DOUBLE)))|
+----+-----------------------------------------------------------------------------------------+
|2003|                                                                                      0.0|
|2007|                                                                     0.22865853658536586|
|2010|                                                                                      0.0|
|2006|                                                                                      0.0|
|2004|                                                                                      0.0|
|2005|                                                                                      0.0|
|2009|                                                                                      0.0|
|2008|                                                                                      0.0|
+----+-----------------------------------------------------------------------------------------+

Time taken: 714 ms
```

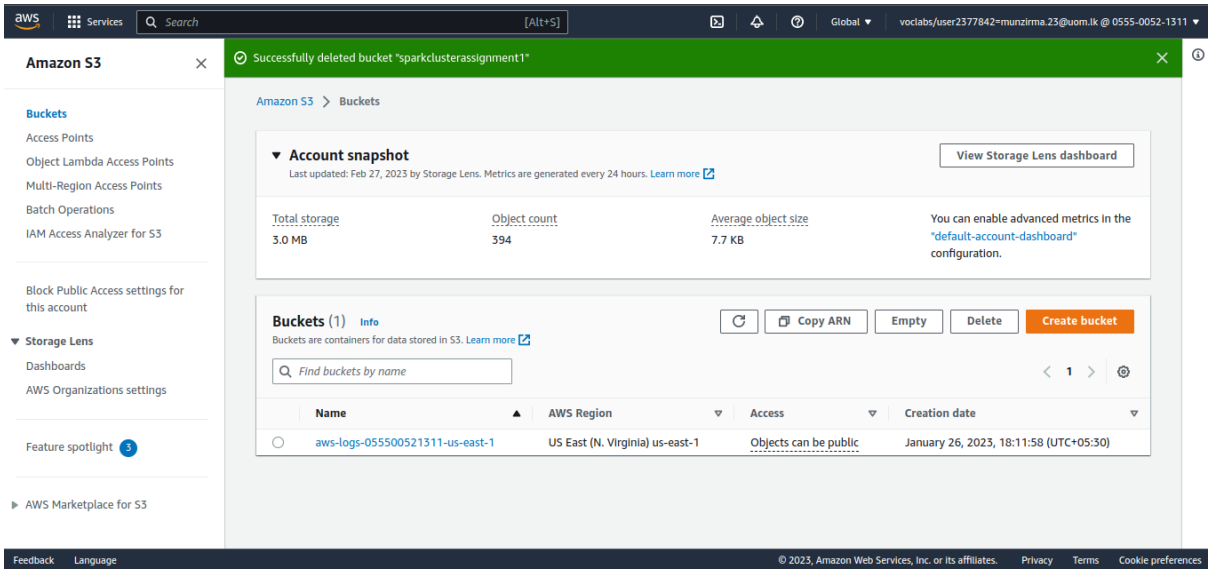
```
scala> spark.time(spark.sql("SELECT Year, avg((SecurityDelay /ArrDelay)*100) from delay_flights WHERE Year is NOT NULL GROUP BY Year").show());
+----+-----------------------------------------------------------------------------------------+
|Year|avg(((CAST(SecurityDelay AS DOUBLE) / CAST(ArrDelay AS DOUBLE)) * CAST(100 AS DOUBLE)))|
+----+-----------------------------------------------------------------------------------------+
|2003|                                                                                      0.0|
|2007|                                                                     0.22865853658536586|
|2010|                                                                                      0.0|
|2006|                                                                                      0.0|
|2004|                                                                                      0.0|
|2005|                                                                                      0.0|
|2009|                                                                                      0.0|
|2008|                                                                                      0.0|
+----+-----------------------------------------------------------------------------------------+

Time taken: 551 ms
```

```
scala> spark.time(spark.sql("SELECT Year, avg((SecurityDelay /ArrDelay)*100) from delay_flights WHERE Year is NOT NULL GROUP BY Year").show());
+----+-----------------------------------------------------------------------------------+
|Year|avg(((CAST(SecurityDelay AS DOUBLE) / CAST(ArrDelay AS DOUBLE)) * CAST(100 AS DOUBLE)))|
+----+-----------------------------------------------------------------------------------+
|2003|                                                                                0.0|
|2007|                                                                  0.22865853658536586|
|2010|                                                                                0.0|
|2006|                                                                                0.0|
|2004|                                                                                0.0|
|2005|                                                                                0.0|
|2009|                                                                                0.0|
|2008|                                                                                0.0|
+----+-----------------------------------------------------------------------------------+

Time taken: 755 ms

scala> 
```

9. Terminating the Cluster and cleaning the S3 bucket




aws
Services
Search
[Alt+S]
Global
voclabs/user2377842=munzirma.23@uom.lk @ 0555-0052-1311
Amazon S3
Successfully deleted bucket "sparkclusterassignment1"
Buckets
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
AWS Organizations settings
Feature spotlight 3
AWS Marketplace for S3
Amazon S3 > Buckets
Account snapshot
Last updated: Feb 27, 2023 by Storage Lens. Metrics are generated every 24 hours. Learn more
View Storage Lens dashboard
Total storage
3.0 MB
Object count
394
Average object size
7.7 KB
You can enable advanced metrics in the "default-account-dashboard" configuration.
Buckets (1) Info
Buckets are containers for data stored in S3. Learn more
Copy ARN
Empty
Delete
Create bucket
Find buckets by name
1
Name
AWS Region
Access
Creation date
aws-logs-055500521311-us-east-1
US East (N. Virginia) us-east-1
Objects can be public
January 26, 2023, 18:11:58 (UTC+05:30)
Feedback
Language
© 2023, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences